5-2019

thornado-hydro: Generalizing discontinuous Galerkin methods for a nuclear equation of state for supernova hydrodynamics

Brandon Lynn Barker
University of Tennessee, Knoxville, bbarker5@vols.utk.edu

Eirik Endeve
University of Tennessee, Knoxville, endevee@ornl.gov

Anthony Mezzacappa
University of Tennessee, Knoxville, mezzacappaa@ornl.gov

DRAFT VERSION MAY 6, 2019
Typeset using LaTeX default style in AASTeX62

thornado-hydro: Generalizing discontinuous Galerkin methods for a nuclear equation of state for supernova hydrodynamics

BRANDON BARKER,[1] EIRIK ENDEVE,[1,2,3] AND ANTHONY MEZZACAPPA[1,3]

[1]*Department of Physics and Astronomy, University of Tennessee Knoxville, TN 37996*
[2]*Computer Science and Mathematics Division, Oak Ridge National Laboratory, TN 37831*
[3]*Joint Institute for Computational Sciences, Oak Ridge National Laboratory, TN 37831*

(Received May 6, 2019)

ABSTRACT

A problem of high importance in computational astrophysics is obtaining accurate solutions to the Euler equations of hydrodynamics. We are interested in solving the Euler equations in the context of core collapse supernovae. The toolkit for high-order neutrino-radiation hydrodynamics (thornado) is being developed for core-collapse supernova (CCSN) simulations and related problems utilizing a spatial discretization based on the discontinuous Galerkin (DG) method. The Euler equations form a hyperbolic set of partial differential equations. In the quasi-linear form, the system can be represented as a set of independent advection equations that can be limited separately. This use of characteristic variables also increases the efficiency of the slope limiting process for high-order DG methods. However, the addition of a tabulated nuclear equation of state to the Euler equations makes the decomposition nontrivial. We introduce the framework for the characteristic decomposition of the Euler equations with the inclusion of the nuclear EOS terms and present results from some initial tests using a third-order scheme. The results confirm that performing limiting on the characteristic variables provides better numerical solutions and is perhaps suited to applications in CCSN simulations.

Keywords: supernovae: general – hydrodynamics – equation of state – methods: numerical – discontinuous Galerkin

1. INTRODUCTION

The core collapse supernovae (CCSNe) explosion mechanism is fundamentally three-dimensional in nature (see e.g., Blondin & Mezzacappa 2006; Müller et al. 2012; O'Connor & Couch 2018). Alongside general reletavistic gravity, complex nuclear matter equations of state (EOS), and neutrino transport, hydrodynamics must be accurately modelled, creating a very challenging and compelling problem. Hydrodynamic instabilities are critical in aiding the explosion, with complex phenomena such as turbulence and convection playing key roles in the CCSN mechanism (Murphy & Meakin 2011; Murphy et al. 2013; Couch et al. 2013; Couch & Ott 2015; Radice et al. 2016; Mabanta & Murphy 2018). The accurate and efficient modelling of the supernova hydrodynamics is imperative if the explosion is to be realistically modelled. For in-depth reviews of the CCSN mechanism, see Bethe (1990); Janka et al. (2007, 2012, 2016); Burrows (2013); Hix et al. (2014); Müller et al. (2016); Couch (2017).

Discontinuous Galerkin (DG) methods have been shown by others (see e.g., Radice & Rezzolla 2011; Schaal et al. 2015; Zanotti et al. 2015; Dumbser et al. 2018) to have a high potential for applications to astrophysical problems. They are a desireable choice for modelling fluid flows in CCSNe. These methods combine elements of spectral and finite volume methods, and achieve high-order accuracy on a compact stencil.

The **t**oolkit for **h**igh-**or**der **n**eutrino-r**ad**iation hydr**o**dynamics (thornado)[1] is being developed to simulate neutrino-radiation hydrodynamics in CCSNe and related applications in nuclear astrophysics. The spatial discretization of solvers for hyperbolic partial differential equations in thornado is based on the DG method. Whether the high-order approach will improve accuracy and efficiency of CCSN models remains to be demonstrated. In this paper we provide an initial description and encouraging demonstration of the DG method implemented in thornado to solve the non-relativistic Euler equations. We focus on basic one-dimensional tests focusing on the implementation of limiters in thornado.

[1] https://github.com/endeve/thornado

The DG slope limiting techniques implemented in `thornado` are similar to those used by Schaal et al. (2015), but is supplemented with a troubled-cell indicator (Fu & Shu 2017) to prevent excessive limiting (e.g., around smooth extrema). It has been shown (see e.g., Schaal et al. 2015; Cockburn & Shu 1998; Cockburn et al. 1989) that limiting on the chracteristic variables (characteristic limiting), which represent advected quantities, shows an improvement over limiting directly on the conserved variables (componentwise limiting). In this paper, we develop a chracteristic limiter for use with a general tabulated nuclear matter EOS in `thornado`. In the case of the ideal EOS, the eigenvectors and eigenvalues which diagonalize the flux Jacobian, which are necessary to transform to the characteristic variables, are known analytically. All of the thermodynamic derivatives have analytic forms. The inclusion of a tabulated nuclear EOS complicates matters, as derivatives must be appriximated by interpolating from the EOS table and the diagonalization becomes non-trovial. We provide explicit analytic expressions for the matrices which diagonalize the flux Jacobian of the Euler equations.

This paper is organized as follows: in Section 2 we briefly describe the Euler equations of gas dynamics including a conservation law appropriate for a nuclear EOS. Section 3 describes our numerical implementation of the DG method in the one dimensional (1D) case. Section 4 presents the eigenvectors used in the diagonalization of the flux Jacobian of the Euler equations, and Section 5 shows the results of preliminary 1D hydrodynamic Riemann problem similar to that of Sod (Sod 1978).

2. EULER EQUATIONS OF GAS DYNAMICS IN CARTESIAN COORDINATES

The non-relativistic Euler equations of gas dynamics (see, e.g., LeVeque 2002, for details in the case of an ideal EOS) in the absence of sources with a nuclear matter EOS are given by the equations of conservation of mass,

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho\, \mathbf{v}) = 0 \tag{1}$$

conservation of momentum,

$$\partial_t (\rho\, \mathbf{v}) + \boldsymbol{\nabla} \cdot (\rho\, \mathbf{v} \otimes \mathbf{v} + P\, \mathbf{I}) = 0 \tag{2}$$

conservation of energy,

$$\partial_t E + \boldsymbol{\nabla} \cdot [(E + P)\, \mathbf{v}] = 0 \tag{3}$$

and conservation of electrons

$$\partial_t D_e + \boldsymbol{\nabla} \cdot (D_e\, \mathbf{v}) = 0 \tag{4}$$

where ρ represents mass density, $\mathbf{v}$ the fluid velocity vector, P the fluid pressure, $D_e = \rho y_e$ where y_e is the electron fraction, $E = \epsilon\rho + \frac{1}{2}\rho v^2$ the total energy (internal plus kinetic), ϵ is the specific internal energy, and $\mathbf{I}$ is the identity tensor. (1)-(4). The inclusion of Equation (4) is because we require a nuclear EOS. These equations are closed by a tabulated EOS where the pressure is given by a function of density, temperature T and the electron fraction: $P = P(\rho, T, y_e)$. We may rewrite Equations (1)-(4) in a more convenient way:

$$\partial_t \mathbf{U} + \boldsymbol{\nabla} \cdot \mathbf{F}(\mathbf{U}) = 0, \tag{5}$$

where $\mathbf{U} = (\rho, \rho\mathbf{v}, E, D_e)^T$ is the vector of conserved quantities and $\mathbf{F}(\mathbf{U}) = (\rho\mathbf{v}, \rho\,\mathbf{v} \otimes \mathbf{v} + P\,\mathbf{I}, (E + P)\mathbf{v}, D_e\mathbf{v})^T$ is the flux vector.

3. NUMERICAL IMPLEMENTATION

3.1. *The Discontinuous Galerkin Methods*

In our solver we have chosen the discontinuous Galerkin (DG) method for our spatial discretization (see e.g., Schaal & Springel 2015; Zhang & Shu 2010; Cockburn & Shu 1998; Cockburn et al. 1989). In this section we will briefly discuss our implementation of the DG method, introducing notation and concepts. For simplicity, we will focus on the one dimensional (1D) case. Recall that we seek solutions to the Euler equations of hydrodynamics, constituting a hyperbolic conservation law of the form

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = 0 \tag{6}$$

Where $\mathbf{U}$ is the evolved state vector and $\mathbf{F}(\mathbf{U})$ is the flux. In order to solve Equation (6) numerically, we divide the computational domain $D \subset \mathbb{R}$ into a disjoint union $\mathcal{T}$ of open elements $\boldsymbol{K}$ such that $D = \cup_{\boldsymbol{K}\in\mathcal{T}} \boldsymbol{K}$. Each element $\boldsymbol{K}$ is a box in the coordinates

$$\boldsymbol{K} \in (x_{\text{L}}, x_R), \tag{7}$$

where x_L and x_R are the left and right boundaries of the cell. We let the approximation space $\mathbb{V}^k$ for the DG method be polynomials of maximal degree k. Note that functions in $\mathbb{V}^k$ can be discontinuous across element interfaces. The DG problem is then to find $\mathbf{U}_h \in \mathbb{V}^k$ which approximates $\mathbf{U}$ in Equation (6), such that $\forall\ \phi \in \mathbb{V}^k$ and $\boldsymbol{K} \in \mathcal{T}$

$$\partial_t \int_{\boldsymbol{K}} \mathbf{U}_h\, \phi\, dx + \int_{\boldsymbol{K}} \partial_x\, \mathbf{F}(\mathbf{U})\, \phi\, dx = 0. \tag{8}$$

Integrating the second term by parts, this becomes

$$\partial_t \int_{\boldsymbol{K}} \mathbf{U_h}\, \phi\, dx + \widehat{\mathbf{F}}(\mathbf{U}_h)\, \phi^-\big|_{x_R} + \widehat{\mathbf{F}}(\mathbf{U}_h)\, \phi^+\big|_{x_L} + \int_{\boldsymbol{K}} \mathbf{F}(\mathbf{U_h})\, \partial_x \phi\, dx = 0. \tag{9}$$

In Eq. (9), $\widehat{\boldsymbol{F}}(\boldsymbol{U}_h)$ is a numerical flux approximating the flux on the boundary of $\boldsymbol{K}$. The numerical flux function is evaluated using values from both boundaries of an element; i.e.,

$$\widehat{\boldsymbol{F}}(\boldsymbol{U}_h) = \boldsymbol{f}(\boldsymbol{U}_h(x^-), \boldsymbol{U}_h(x^+)), \tag{10}$$

where superscripts $-/+$ indicate that the function is evaluated to the immediate left/right of the interface. We use the Harten-Lax-van Leer (HLL) flux (Harten et al. 1983) for all the numerical experiments presented in Section 5.

In each element $\boldsymbol{K}$, we use a nodal representation of the conserved variables $\mathbf{U}$:

$$\boldsymbol{U}(x,t) \approx \boldsymbol{U}_h(x,t) = \sum_{i=1}^{N=k+1} \boldsymbol{U}_i(t)\, \ell_i(x), \quad \text{where} \quad \ell_i(\eta) = \prod_{\substack{j=1 \\ j\neq i}}^{N=k+1} \frac{\eta - \eta_j}{\eta_i - \eta_j} \tag{11}$$

are Lagrange polynomials defined on $I = \{\eta : \eta \in (-0.5, 0.5)\}$, and are constructed to interpolate the node set $S_N = \{\eta_i\}_{i=1}^N \subset I$. The spatial coordinate x and the reference coordinate η are related by the mapping $x(\eta) = x_L + (0.5 + \eta)\, \Delta x$. Then, for any $\eta_j \in S_N$, $\ell_i(\eta_j) = \delta_{ij}$, so that $\boldsymbol{U}_h(x(\eta_j), t) = \boldsymbol{U}_j(t)$. We define the M-point quadrature $Q_M : C^0(I) \to \mathbb{R}$ with abscissas $\hat{S}_M = \{\eta_q\}_{q=1}^M$ and weights $\{w_q\}_{q=1}^M$, normalized such that $\sum_{q=1}^M w_q = 1$ in order to evaluate the integrals in Eq. (9). We use the M-point Legendre-Gauss quadrature, which is exact for polynomials of degree $\leq 2M - 1$. Then, if $P_h(x)$ is such a polynomial, we have

$$\frac{1}{\Delta x} \int_{\boldsymbol{K}} P_h(x)\, dx = \int_I P_h(\eta)\, d\eta = \sum_{q=1}^M w_q\, P_h(\eta_q). \tag{12}$$

We let $M = N$ and $S_N = \hat{S}_N$, which is a spectral-type nodal collocation DG approximation (Bassi et al. 2013), which is exact for Cartesian coordinates and Legendre-Gauss quadrature. Inserting Eq. (11) into Eq. (9), letting $\phi(x) = \ell_k(x)$, using the quadratures defined above, we obtain

$$\partial_t \int_{\boldsymbol{K}} \boldsymbol{U}_h\, \phi\, dx = w_k\, \partial_t \boldsymbol{U}_k\, \Delta x \tag{13}$$

for the time derivative piece, where $\Delta x = x_R - x_L$. Similarly, the last term on the left-hand side of Eq. (9) becomes

$$\int_K \boldsymbol{F}(\boldsymbol{U}_h)\, \frac{\partial \phi}{\partial x}\, dx = \sum_{q=1}^N w_q\, \boldsymbol{F}(\boldsymbol{U}_q)\, \frac{\partial \ell_k}{\partial \eta}(\eta_q). \tag{14}$$

Now we may combine Eqs. (13)-(14) in Eq. (9) resulting in the semi-discrete form

$$\frac{d\boldsymbol{U}_k}{dt} = -\frac{1}{w_k \Delta x} \Big\{ \Big[\widehat{\boldsymbol{F}} \ell_k\big|_{x_R} - \widehat{\boldsymbol{F}} \ell_k\big|_{x_L} \Big] - \sum_{q=1}^N w_q\, \boldsymbol{F}(\boldsymbol{U}_q)\, \frac{\partial \ell_k}{\partial \eta}(\eta_q) \Big\}. \tag{15}$$

We now have a system of ordinary differential equations (ODEs), which may be evolved in time with an ODE solver. In Section 5 we use the third-order strong stability-preserving Runge-Kutta (SSP-RK3) method (Shu & Osher 1988).

3.2. *Slope Limiting*

A common feature of high-order numerical PDEs is unphysical oscillations in the solutions around discontinuities. It is therfore of great interest in the DG algorithm to implement slope limiting of the polynomial $\mathbf{U}_h$. We use the total variation diminishing (TVD) slope limiter (see, e.g., Cockburn & Shu 1998) in conjunction with the troubled cell indicator discussed in Fu & Shu (2017) to prevent excessive limiting by flagging elements where limiting is needed. To do this, we need to reduce under- and overshootings of the higher-order solution at cell boundaries compared to the cell averages of neighbor cells. Recall from Eq. (11) that in each cell our solution is expressed in the nodal form. It is convenient, however, for limiting purposes to express the solution in a modal representation:

$$\mathbf{U}_h(x,t) = \sum_{l=1}^{N=k+1} c_l(t)\, P_l(x) \tag{16}$$

where $P_l(x)$ are the Legendre polynomials. These representations of the solution are equivalent by requiring weak equivalence

$$\sum_{j=1}^{N} \int_{\boldsymbol{K}} \left(\boldsymbol{U}_j(t)\, \ell_j(x) - c_j(t)\, P_j(x)\right) \phi \, dx = 0 \quad \forall\, \phi \in \mathbb{V}^k. \tag{17}$$

Letting ϕ be the Lagrange interpolating polynomials ℓ_i, we have

$$\sum_{j=1}^{N} \int_{\boldsymbol{K}} \ell_i(x)\, \ell_j(x)\, dx\, \boldsymbol{U}_j(t) = \sum_{j=1}^{N} \int_{\boldsymbol{K}} \ell_i(x)\, P_j(x)\, dx\, c_j(t). \tag{18}$$

Now letting $M_{ij} = \int_{\boldsymbol{K}} \ell_i(x)\, \ell_j(x)\, dx$ be the mass matrix, $A_{ij} = \int_{\boldsymbol{K}} \ell_i(x)\, P_j(x)\, dx$, $\bar{\boldsymbol{U}} = \{U_1, ..., U_N\}$ be the nodal coefficients, and $\bar{\boldsymbol{c}} = \{c_1, ..., c_N\}$ be the modal coefficients, we see that the different representations are related by a linear transformation

$$M\bar{\boldsymbol{U}} = A\bar{\boldsymbol{c}} \tag{19}$$

To perform slope limiting, we compare the weight c_2, which is proportional to the first derivative of the solution in the cell, to the neighboring cell averages by the following

$$\mathcal{M}\,\widetilde{c}_2 = \text{minmod}(\mathcal{M}c_2, \beta_{\text{TVD}}\mathcal{M}(c_1^+ - c_1), \beta_{\text{TVD}}\mathcal{M}(c_1 - c_1^-)) \tag{20}$$

where $\widetilde{c}_2$ is the limited weight. The superscripts $-/+$ indicate the cell averages of the neighboring cells to the immediate left/right of the interface and $\mathcal{M}$ is a transformation matrix. For componentwise limiting, we let $\mathcal{M}$ be the identity matrix. Notice that

$$\frac{1}{\Delta x}\int_{\boldsymbol{K}} \boldsymbol{U}_h\, dx = \frac{1}{\Delta x}\sum_{l=1}^{N} \int_{\boldsymbol{K}} P_l(x)\, \cdot 1\, dx\, c_l = \delta_{l1} c_l = c_1 \tag{21}$$

so c_1 is precisely the cell average. The minmod function in Eq. (20) is defined as

$$\text{minmod}(a_1, a_2, a_3) = \begin{cases} s\ \min\{|a_1|, |a_2|, |a_3|\} & s = \text{sign}(a_1) = \text{sign}(a_2) = \text{sign}(a_3) \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

The parameter β_{TVD} takes values in the closed interval $[1, 2]$ and scales the strength of the limiting. A minimal β_{TVD} corresponds with the total variation diminishing scheme but is more dissipative than the maximal β_{TVD} case, which is potentially more oscillatory. Increasing β_{TVD} puts more weight on the neighboring cell averages, making the minmod function more likely to set $\widetilde{c}_2 = c_2$, when no limiting is applied. If instead $\widetilde{c}_2 \neq c_2$, the solution is truncated to first order

$$\boldsymbol{U}_h \to \widetilde{\boldsymbol{U}}_h = c_1\, P_1(x) + \widetilde{c}_2\, P_2(x) \tag{23}$$

and, by Eq. (22), $\widetilde{c}_2$ can potentially be zero as well. Notice that, by Eq. (21), the cell average is not altered by the limiting process, so the process is conservatory.

In order to determine where slope limiting is necessary, we use the troubled cell indicator (TCI) (Fu & Shu 2017) to prevent excessive limiting

$$I_{\boldsymbol{K}}(G) = \frac{\sum_j |G_{\boldsymbol{K}} - G_{\boldsymbol{K}}^{(j)}|}{\max_j |G_{\boldsymbol{K}^{(j)}}^{(j)}|}, \tag{24}$$

where $G \in \boldsymbol{G} \subseteq \boldsymbol{U}$ and $\boldsymbol{G} = (\rho, E)$. Here, the sum in the numerator and the max in the denominator are taken over the neighboring elements sharing a boundary with target element $\boldsymbol{K}$, $G_{\boldsymbol{K}}$ is the cell average in $\boldsymbol{K}$, $G_{\boldsymbol{K}}^{(j)}$ is the cell average computed by extrapolating the polynomial representation from the neighboring element $\boldsymbol{K}^{(j)}$ into $\boldsymbol{K}$, and $G_{\boldsymbol{K}^{(j)}}^{(j)}$ is the cell average native to neighbor element $\boldsymbol{K}^{(j)}$. An element is flagged for limiting if, for any $G \in \boldsymbol{G}$, $I_{\boldsymbol{K}}(G) > C_{\text{TCI}}$, where C_{TCI} is a defined threshold.

3.3. *Time Integration*

Eq. (15) presented a system of ODEs that must be evolved in time. We use the third-order strong stability-preserving Runge-Kutta (SSP-RK) scheme (Shu & Osher 1988). The general s-stage Runge-Kutta time stepping algorithm, including the limiting process, can be summarized as in Cockburn (2001):

1. Set $\bar{\boldsymbol{U}}^{(0)} = \bar{\boldsymbol{U}}^{n}$,
2. For $i = 1, \ldots, s$ compute:
$$\bar{\boldsymbol{U}}^{(i)} = \Lambda^{\text{TVD}}\Big\{ \sum_{j=0}^{i-1} \alpha_{ij}\, \bar{\boldsymbol{U}}^{(j)} + \beta_{ij}\, \Delta t\, \bar{\boldsymbol{F}}\big(\bar{\boldsymbol{U}}^{(j)}\big) \Big\}, \tag{25}$$
3. Set $\bar{\boldsymbol{U}}^{n+1} = \bar{\boldsymbol{U}}^{(s)}$.

Above, the TVD limiter limiter preventing unphysical states is denoted by the operator $\Lambda^{\text{TVD}}\{\}$. The SSP-RK3 coefficients α_{ij} and β_{ij} may be found in Table 2.1 in Cockburn & Shu (2001). For each step in Eq. (25), the TVD limiter is applied to elements flagged by the troubled cell indicator (where $I_{\boldsymbol{K}} > C_{\text{TCI}}$). This algorithm is subject to a timestep stability condition Δt_{TVD} where $|\lambda|$ is the magnitude of the largest eigenvalue of the flux Jacobian. Then (e.g., Cockburn & Shu (2001)):

$$\Delta t_{\text{TVD}} \leq \frac{1}{d} \frac{(\Delta x/|\lambda|)}{(2k+1)} \tag{26}$$

where d is the dimension. For $d = 1, k = 2$ (third-order scheme) the effective Courant-Friedrichs-Lewy (CFL) factor is 0.2. In the numerical experiments presented in Section 5 we use a more conservative timestep restriction of $\Delta t = 0.1(\Delta x/|\lambda|)$.

3.4. *Characteristic Decomposition*

Experience has shown that the slope limiting described in the previous section is more effecient when performed on the so-called 'characteristic variables' as opposed to the conserved variables $\mathbf{U}_h$ (see, e.g., Cockburn & Shu 1998, for a description). Inspired by theory for linear systems, we begin by rewriting Eq. (5) in the quasi-linear form as follows

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial x} = 0. \tag{27}$$

Because the Euler equations form a system of hyperblic partial differential equations (see, e.g., LeVeque 1992), we can decompose the Jacobian of the flux vector as

$$\frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}} = \mathcal{R}\Lambda\mathcal{R}^{-1}, \tag{28}$$

where the columns of $\mathcal{R}$ contain the right eigenvectors of the Jacobian, the rows of $\mathcal{R}^{-1}$ contain the left eigenvectors, and Λ is a diagonal matrix containing the eigenvalues of the Jacobian. For hyperbolic systems, the eigenvalues are real and the eigenvectors form a complete set (see e.g., LeVeque 1992). At this point, we will introduce the characteristic variable $\mathbf{w} = \mathcal{R}^{-1}\mathbf{U}$. Multiplying both sides of Equation (27) by $\mathcal{R}^{-1}$ and simplifying, we are able to linearize the system of equations to a system of advection equations

$$\frac{\partial \mathbf{w}}{\partial t} + \Lambda \frac{\partial \mathbf{w}}{\partial x} = 0. \tag{29}$$

Solutions to these now decoupled advection equations are far simpler to obtain than the previous equations. Recall in Eq. (22) that the limiting cell averages were transformed by a matrix $\mathcal{M}$. If we let $\mathcal{M} = \mathcal{R}^{-1}$ then limiting is performed on the characteristic variables. Limiting may then be applied to dampen oscillations in the solutions for the characteristic variables $\mathbf{w}$, which are then transformed back to the solution of the conserved variables $\mathbf{U}$ (see e.g., Cockburn & Shu 1998; Schaal et al. 2015, for a description). While this process of characteristic limiting has been done for the ideal EOS (Cockburn & Shu 1998), we want to extend this process to the case of a tabulated nuclear matter EOS.

4. JACOBIAN AND EIGENSYSTEM FOR THE NUCLEAR CASE

We assume that pressure $P = P(\tau, \epsilon, D_e)$, where $\tau = \frac{1}{\rho}$. Let the vector of conserved variables be $\mathbf{U} = \{\rho, m_1, m_2, m_3, E, D_e\}$, where $m_i = \rho v_i$ and $D_e = \rho y_e$. The flux vector is $\mathbf{F}(\mathbf{U}) = \{m_1, m_1^2\tau + P, m_1 m_2 \tau, m_1 m_3 \tau, (E+P)m_1\tau, D_e m_1 \tau\}$. The Jacobian of the flux vector is given by

$$\frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -v_1^2 - P_\tau \tau^2 - P_\epsilon \tau(\epsilon - \frac{v^2}{2}) & v_1(2 - P_\epsilon \tau) & -P_\epsilon v_2 \tau & -P_\epsilon v_3 \tau & P_\epsilon \tau & P_{D_e} \\ -v_1 v_2 & v_2 & v_1 & 0 & 0 & 0 \\ -v_1 v_3 & v_3 & 0 & v_1 & 0 & 0 \\ v_1(-H - P_\tau \tau^2 - P_\epsilon \tau(\epsilon - \frac{v^2}{2})) & H - P_\epsilon v_1^2 \tau & -P_\epsilon v_1 v_2 \tau & -P_\epsilon v_1 v_3 \tau & v_1(1 + P_\epsilon \tau) & v_1 P_{D_e} \\ -v_1 y_e & y_e & 0 & 0 & 0 & v_1 \end{bmatrix} \tag{30}$$

Where $H = (E+P)\tau$ is the specific enthalpy and

$$P_\epsilon = \left(\frac{\partial P}{\partial \epsilon}\right)_{\tau, D_e}, \quad P_{D_e} = \left(\frac{\partial P}{\partial D_e}\right)_{\tau, \epsilon}, \quad P_\tau = \left(\frac{\partial P}{\partial \tau}\right)_{\epsilon, D_e} \tag{31}$$

are the necessary thermodynamic derivatives. Expressions for these derivatives in terms of the table variables $\rho, T,$ and y_e are given in Appendix A. The eigenvalues of the flux Jacobian are given by the diagonal matrix

$$\Lambda = \begin{bmatrix} v_1 - c_s & 0 & 0 & 0 & 0 & 0 \\ 0 & v_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & v_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & v_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & v_1 + c_s \end{bmatrix} \tag{32}$$

where $c_s = \sqrt{\Gamma P \tau}$, with $\Gamma = \left(\tau(P P_\epsilon - P_\tau) + P_{D_e} y_e \tau^{-1}\right) P^{-1}$, is the local sound speed. In the case of the general EOS $P = P(\tau, \epsilon)$, this reduces to the sound speeds given in Colella & Glaz (1985). The right eigenvectors are then given by the column vectors of the following matrix

$$\mathcal{R}_1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ v_1 - c & 0 & v_1 & v_1 & 0 & c + v_1 \\ v_2 & 1 & 0 & 0 & 0 & v_2 \\ v_3 & 0 & 0 & 0 & 1 & v_3 \\ h - c v_1 & v_2 & \beta & 0 & v_3 & h + c v_1 \\ y_e & 0 & 0 & \frac{\tau \chi}{2 P_{D_e}} & 0 & y_e \end{bmatrix}$$

where the following definitions have been used: $h_n = \frac{c^2}{P_\epsilon \tau} + k$, $k = \frac{-y_e P_{D_e} \tau^{-1} + P_\epsilon(\frac{1}{2}v^2 + \epsilon) + P_\tau \tau}{P_\epsilon}$, $\delta_1 = v_1^2 - v_2^2 - v_3^2$, $\chi = P_\epsilon(\delta_1 + 2\epsilon) + 2P_\tau \tau$, and $\beta = \frac{1}{2}(\delta_1 + 2\epsilon + \frac{2P_\tau \tau}{P_\epsilon})$. The left eigenvectors are given by the row vectors of the inverse matrix $\mathcal{L}_1 = \mathcal{R}_1^{-1}$

$$\mathcal{R}_1^{-1} = \frac{1}{c^2}\begin{bmatrix} \frac{1}{4}(2cv_1+\omega) & \frac{1}{2}(-c-\phi_1) & -\frac{1}{2}\phi_2 & -\frac{1}{2}\phi_3 & \frac{P_\epsilon\tau}{2} & \frac{P_{D_e}}{2} \\ -\frac{v_2\omega}{2} & \phi_1 v_2 & c^2+\phi_2 v_2 & \phi_3 v_2 & -\phi_2 & -P_{D_e} v_2 \\ \frac{2\chi c^2+\alpha\omega\tau^{-1}}{2\chi} & -\frac{\phi_1\alpha}{\chi\tau} & -\frac{\phi_2\alpha}{\chi\tau} & -\frac{\phi_3\alpha}{\chi\tau} & \frac{P_\epsilon\alpha}{\chi} & \frac{P_{D_e}\left(\alpha-2c^2\right)}{\tau\chi} \\ -\frac{y_e P_{D_e}\omega}{\chi\tau} & \frac{2y_e P_{D_e}\phi_1}{\chi\tau} & \frac{2y_e P_{D_e}\phi_2}{\chi\tau} & \frac{2y_e P_{D_e}\phi_3}{\chi\tau} & -\frac{2y_e P_{D_e} P_\epsilon}{\chi} & \frac{2P_{D_e}\left(c^2-y_e P_{D_e}\right)}{\tau\chi} \\ -\frac{v_3\omega}{2} & \phi_1 v_3 & \phi_2 v_3 & c^2+\phi_3 v_3 & -\phi_3 & -P_{D_e} v_3 \\ \frac{1}{4}(\omega-2cv_1) & \frac{1}{2}(c-\phi_1) & -\frac{1}{2}\phi_2 & -\frac{1}{2}\phi_3 & \frac{P_\epsilon\tau}{2} & \frac{P_{D_e}}{2} \end{bmatrix}$$

where $\phi_i = P_\epsilon\,\tau\,v_i$, $\omega = \tau\left(P_\epsilon\left(v^2-2\epsilon\right)-2\,P_\tau\,\tau\right)$, and $\alpha = 2y_e P_{D_e} - \tau\chi$. The left and right eigenvectors presented here are similar in form to those presented in Schaal et al. (2015), but are considerably more complicated due to the restriction of a nuclear EOS. Having an explicit form for these matrices is desirable as it eliminates the need to numerically diagonalize the flux Jacobian[2].

5. PRELIMINARY NUMERICAL RESULTS

In this section we present preliminary numerical results obtained with the DG method implemented in `thornado` leveraging use of the characteristic decomposition. Unless otherwise noted, we use the SFHo EOS of nuclear matter (Steiner et al. 2013) now commonly used in high-fidelity CCSN simulations due to its consistency with neutron star mass and radius observations. These tests serve to gauge the performance improvement of the DG characteristic decomposition implementation on a set of benchmarks as an initial assessment of its suitability for future CCSN simulations.

5.1. *Riemann Problem*

We present a 1D Riemann problem inspired by the classic 1D Sod shock tube Riemann problem (Sod 1978) in Cartesian coordinates. This is an ideal first test as it is a fairly simple scenario but contains both a contact discontinuity and a shock to test the efficiency of our limiting scheme, though due to the nature of our EOS, an analytic solution does not exist. The computational domain is D = [-5,5] km with a discontinuity initially located at $x = 0$ km separating the left and right states

$$\mathbf{U}_L = (10^{12}\text{ g cm}^{-3}, 0\,, 2.703\text{x}10^{32}\text{ ergs cm}^{-3}, 0.4\text{x}10^{12})^T$$
$$\mathbf{U}_R = (1.25\text{x}10^{11}\text{ g cm}^{-3}, 0\,, 2.822\text{x}10^{31}\text{ ergs cm}^{-3}, 0.375\text{x}10^{11})^T.$$

The test is run until $t = 0.025$ ms with 100 elements using $C_{\text{TCI}} = 0.2$ and $\beta_{\text{TVD}} = 2.0$ with a third-order time integration SSP-RK3 scheme. Results for density, pressure, velocity, and electron fraction are plotted in the upper left, upper right, lower left, and lower right panels of Figure 1, respectively, compared to a reference run using 10000 elements, third-order time integration, and first order spacial discretization. Using the characteristic limiting described above, the DG method captures the features of the solution, resolving well the contact discontinuity, shock, and rarefraction wave located at about $x = 2$ km, $x = 4$ km, and from $x = -3$ km to $x = 0$ km, respectively, without introducing noticeable oscillations in the numerical solutions near the discontinuities. In Figure 2 we plot elements in the xt-plane flagged by the troubled cell indicator for limiting, showing that the troubled cell indocator flagged cells around both the shock and contact discontinuity for limiting and maintained that for the full time of the test.

5.2. *Optimal Limiting Parameters*

In order to determine the optimal limiting parameters C_{TCI} and β_{TVD}, we performed a suite of sixteen simulations of Sod's problem with various limiting parameters with $\beta_{\text{TVD}} \in [1.0, 2.0]$ and $C_{\text{TCI}} \in [0.0, 0.2]$. For each $(\beta_{\text{TVD}}, C_{\text{TCI}})$ pair, we computed the relative error and total variation

$$\varepsilon(\mathbf{U}) = \frac{1}{N}\sum_{\boldsymbol{K}}\left|\frac{\boldsymbol{U}_h^{\text{ref}}(x_q) - \boldsymbol{U}_h(x_q)}{\boldsymbol{U}_h^{\text{ref}}(x_q)}\right| \qquad TV = \sum_k |\bar{U}_{k+1} - \bar{U}_k| \tag{33}$$

in density and electron fraction at $t = 0.025$ms, where $\boldsymbol{U}_h^{\text{ref}}$ is the reference solution, $\bar{\boldsymbol{U}}_{k+1}$ is the cell average is cell k, and x_q are numerical quadrature points. Results are plotted in Figure 3. We find that, as expected, increasing β_{TVD} tends to monotonically

[2] These matrices and their derivations are available in Mathematica notebooks at https://github.com/AstroBarker/dgHydro-nuc_charDecomp

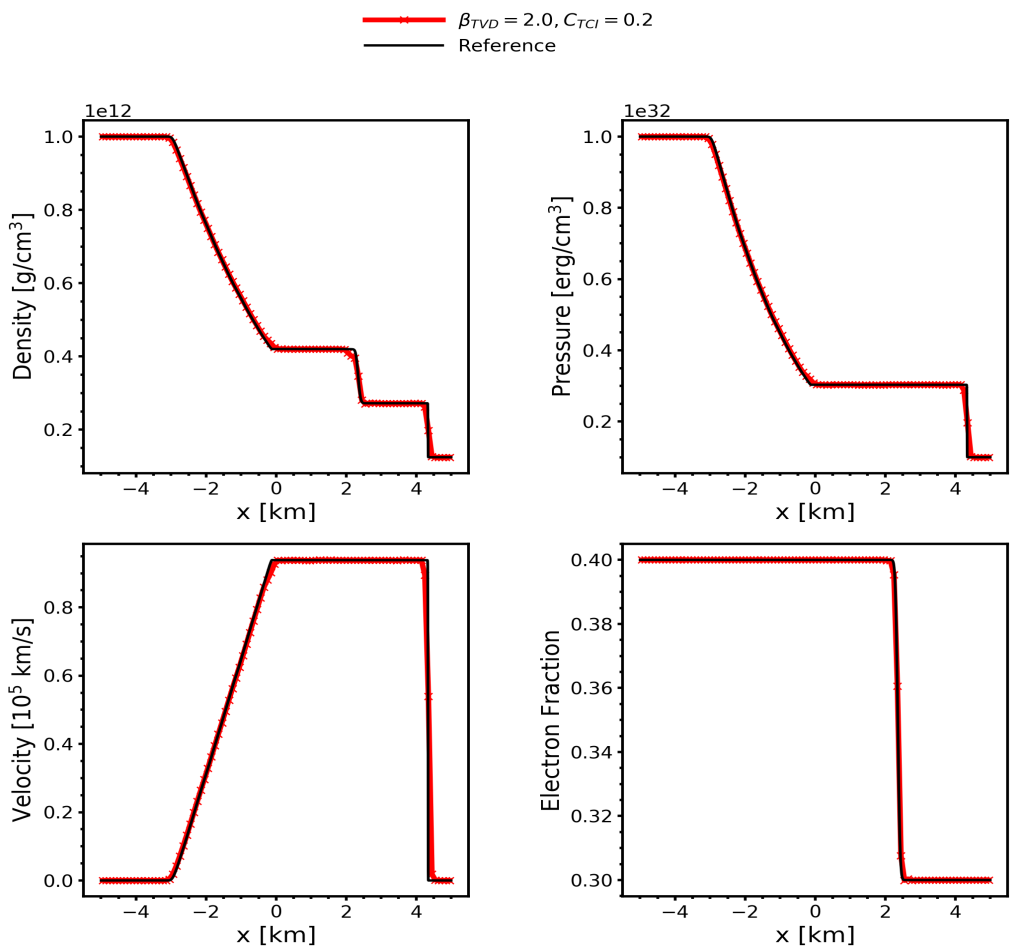

Figure 1. Numerical solution of the Riemann problem using 100 elements and $C_{\text{TCI}} = 0.2$ and $\beta_{\text{TVD}} = 2.0$ with a third-order time integration SSP-RK3 scheme for density (upper left), pressure (upper right), velocity (lower left), and electron fraction (lower right) compared with a reference solution (black) using 10000 elements.

decrease the relative error while (non-monotonically) increasing the total variation due to the limiter allowing for more oscillations. Increasing C_{TCI} tends to decrease the relative error and increase the total variation. We have selected $\beta_{\text{TVD}} = 1.75$ and $C_{\text{TCI}} = 0.1$ to be the optimal parameters providing the best combination of relative error and total variation reduction. Unless otherwise noted, all following results will use this combination of limiting parameters.

5.3. *Improvement From Componentwise Limiting*

The motivation for limiting on the characteristic variables is the potential improvement from componentwise limiting. In Figure 4 we plot the solution at $t = 0.025$ ms using 100 elements employing both characteristic limiting (blue) and componentwise limiting (red) compared to the reference solution computed with 10000 elements. We observe, for the componentwise limiting, noticable oscillations in density around the contact discontinuity as well as less resolved discontinuities. The electron fraction displays very large oscillations at the left side of the contact discontinuity, up to ± 0.02 in the cell averaged electron fraction. The solutions with characteristic limiting are very close to the reference solution, presenting a significant improvement over componentwise limiting. This makes characteristic limiting particularly appealing for CCSN simulations where the electron fraction plays a critical role in the explosion dynamics.

5.4. *High Density Regime*

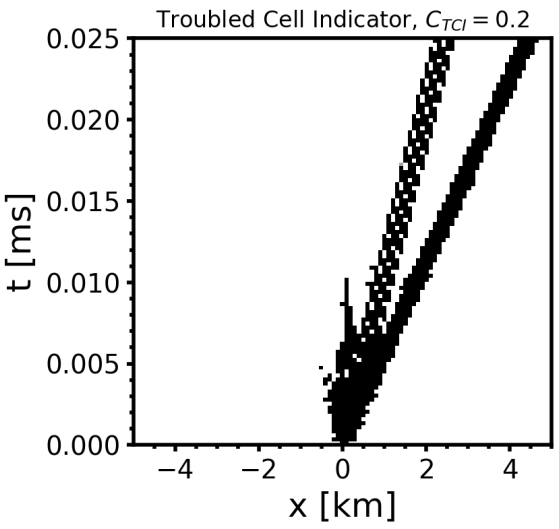

Figure 2. xt-plane of elements flagged for limiting by the troubled cell indicator in the Riemann problem of Section 5.1.

This test is included to verify the performance improvements of characteristic limiting in a higher density regime. It is designed with the Sod problem in mind. The computational domain is $D = [-5, 5]$ km with a discontinuity initially located at $x = 0$ km separating the left and right states

$$\mathbf{U}_L = (10^{13}\text{ g cm}^{-3}, 0\,, 3.712\text{x}10^{32}\text{ ergs cm}^{-3}, 0.15\text{x}10^{12})^T$$
$$\mathbf{U}_R = (1.25\text{x}10^{12}\text{ g cm}^{-3}, 0\,, 3.015\text{x}10^{31}\text{ ergs cm}^{-3}, 0.169\text{x}10^{12})^T.$$

The test is run until $t = 0.05$ ms with 100 elements using $C_{\text{TCI}} = 0.1$ and $\beta_{\text{TVD}} = 1.2$. We chose a lower β_{TVD} here, as larger values resulted in unphysical internal energies outside of the EOS table causing the test to fail, motivating the development of a positivity limiter compatible with a tabulated nuclear matter EOS (Zhang & Shu 2010). Inital state values were chosen to be consistent with high-fidelity CCSN simulation code CHIMERA (Bruenn et al. 2018) data by first choosing the left and right densities to be an order of magnitude larger than in Section 5.1, and then finding temperatures and electron fractions consistent with those densities from CHIMERA data, and computing the other state variables from the EOS table. Results are plotted in Figure 5. Our method captures the main features of the solution, but introduces more

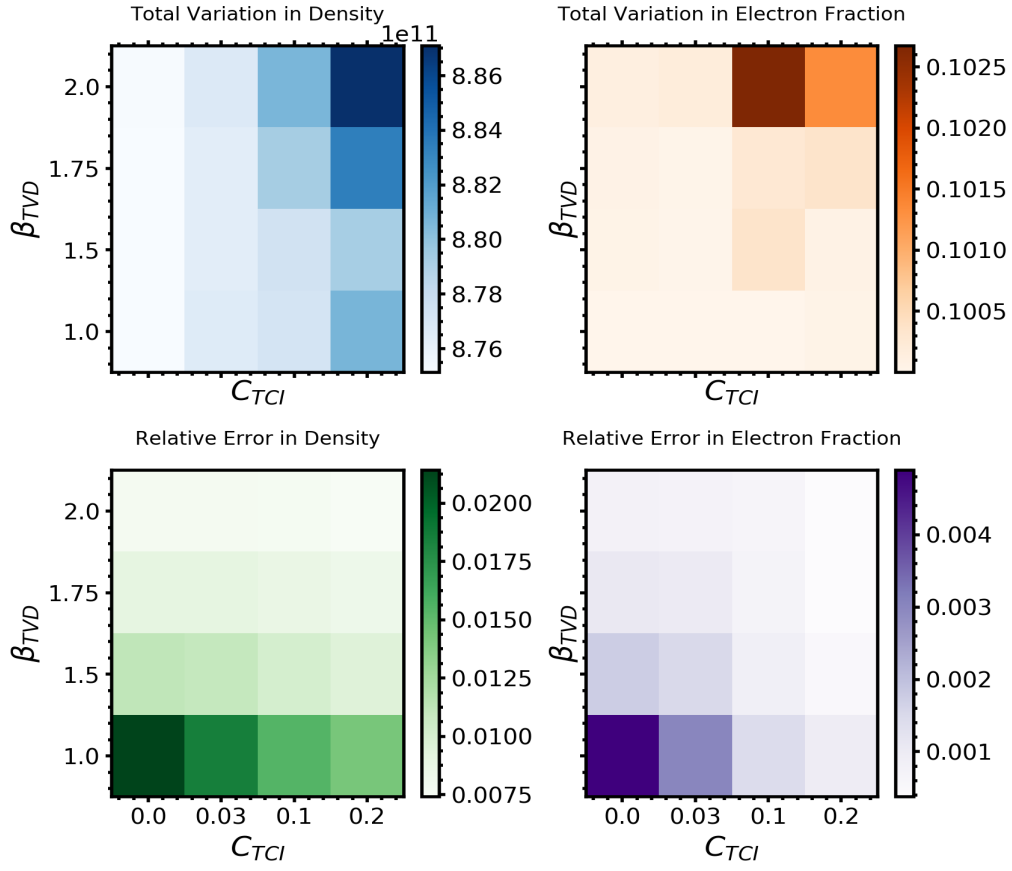

Figure 3. Error and total variation landscapes for various values of the limiting parameters C_{TCI} and β_{TVD}. The top row shows total variation in density (left) and electron fraction (right). The bottom row show relative error in density (left) and electron fraction (right).

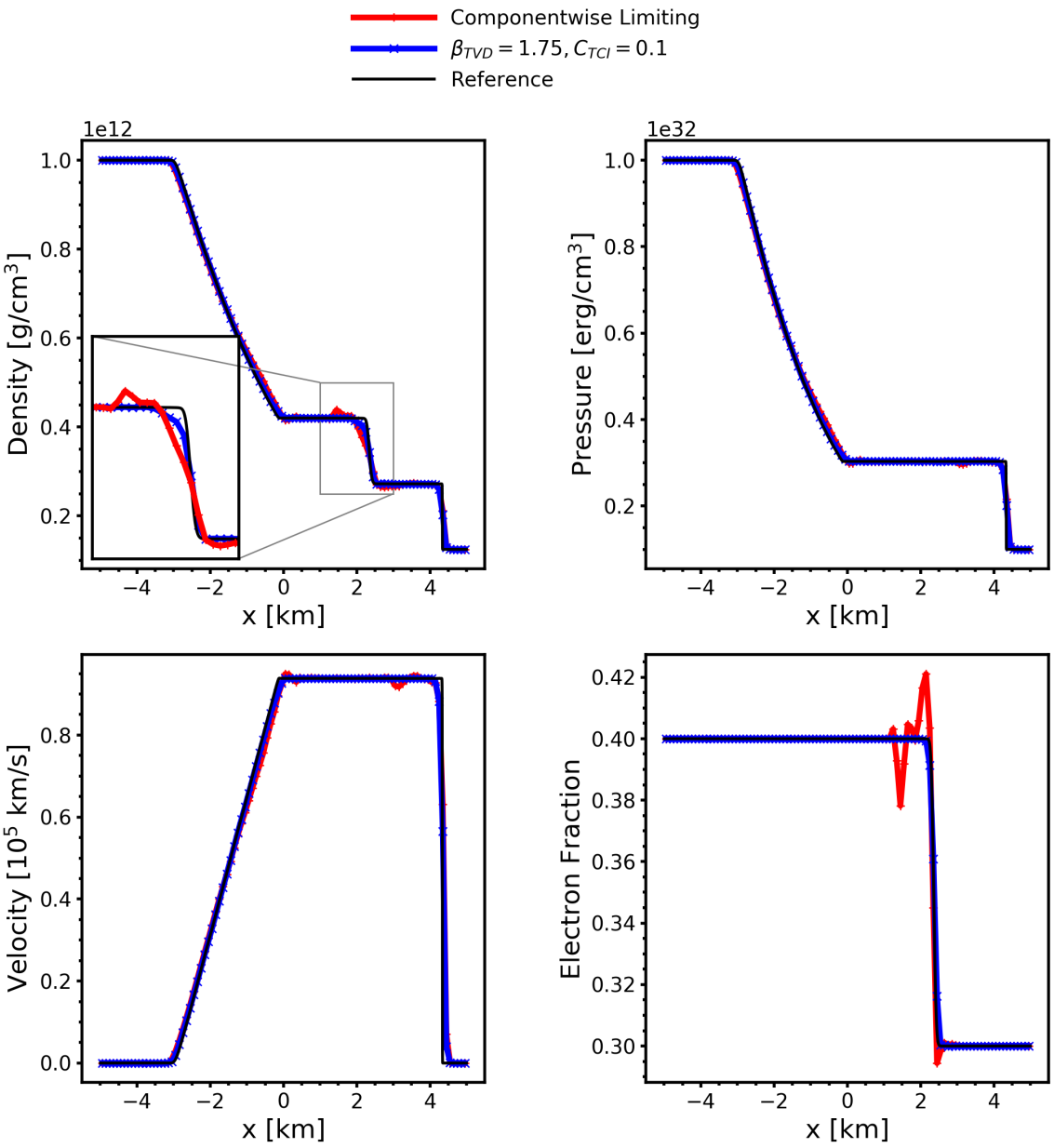

Figure 4. Numerical solution of Sod's problem using 100 elements with characteristic limiting (blue) and componentwise limiting (red) for density (upper left), pressure (upper right), velocity (lower left), and electron fraction (lower right) compared to a reference solution (black) using 10000 elements.compared with a reference solution using 10000 elements.

oscillations than in the lower density case, even when using the characteristic limiter. We notice an undershooting of the right side of the contact discontinuity in the electron fraction not previously present warranting further investigation.

5.5. *EOS Resolution Dependence*

In order to test the sensitivity of the limiter on the EOS table resolution, we repeated the tests in Section 5.3 with a higher resolution SFHo EOS table. As before, tests are computed using 100 elements until $t = 0.025$ms with characteristic limiting and the optimal limiting parameters determined in Section 5.2 and compared to a reference solution computed using 10000 elements. Results for both tables are plotted in Figure 6. We find no sensitivity to the table resolution, with the exception of minor differences in density to the left of the contact discontinuity.

5.6. *EOS Table Sensitivity*

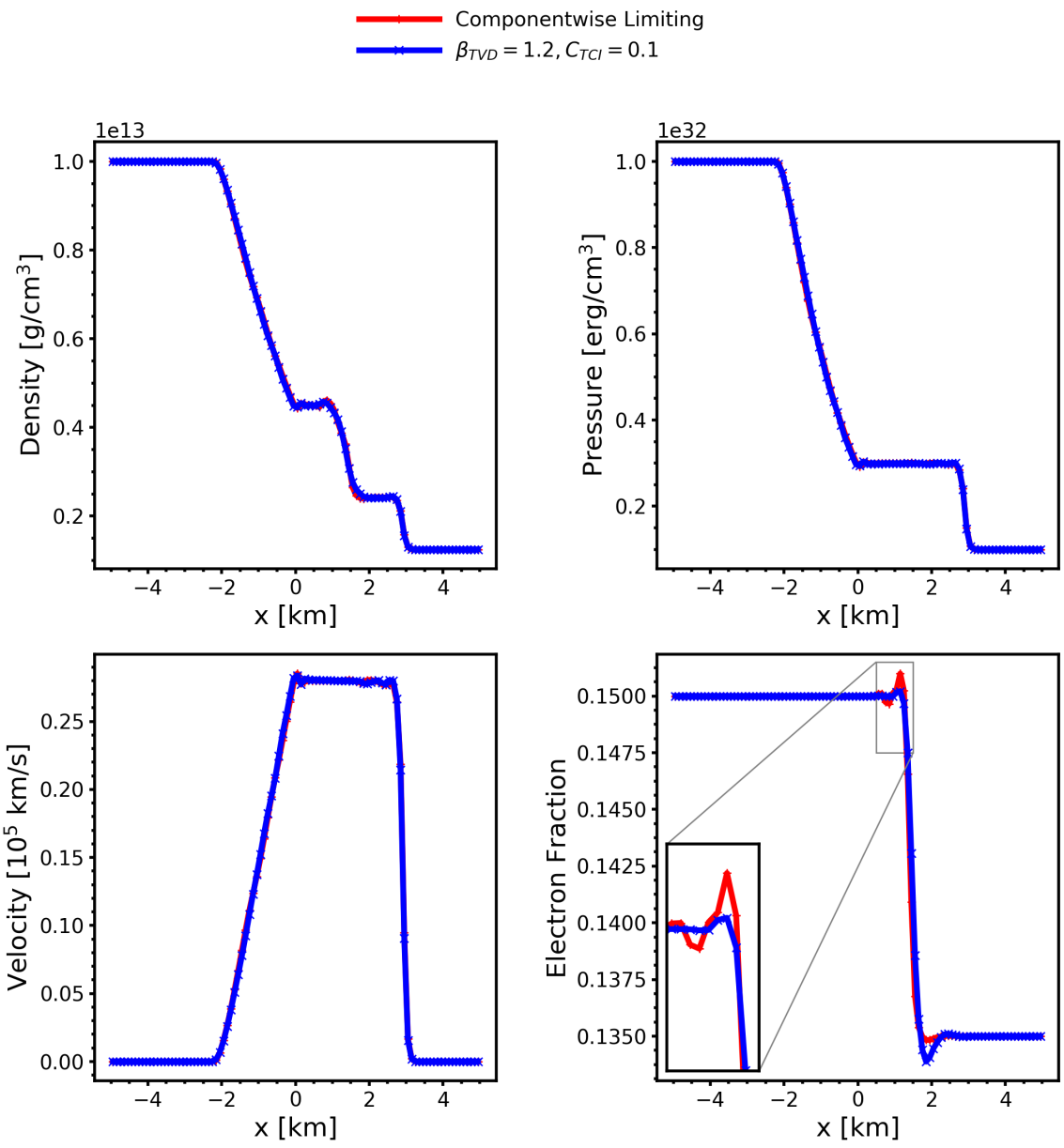

Figure 5. Results for the high density Sod-like problem computed using 100 elements and $C_{\text{TCI}} = 0.1$ and $\beta_{\text{TVD}} = 1.2$ at $t = 0.05$ms.

In order to probe the sensitivity of the method to the EOS table, we repeated the tests conducted in Section 5.3 using three different EOS tables: the high resolution SFHo table from Section 5.5, the SFHx table, and the DD2 table. As before, tests are computed using 100 elements until $t = 0.025$ms with characteristic limiting and the optimal limiting parameters determined in Section 5.2. We find nearly no sensitivity to the EOS table used in the low density regime, with the exception of very small variations in density to the left of the contact discontinuity. Future work includes testing the sensitivity to the EOS table in higher density regimes.

6. CONCLUSIONS

We have presented preliminary developments and numerical results for a characteristic limiter to be used with solvers of the non-relativistic Euler equations of gas dynamics in the **t**oolkit for **h**igh-**or**der **n**eutrino-r**ad**iation hydr**o**dynamics (`thornado`). We presented analytic forms for the diagonalizating matrices for the flux Jacobian of the Euler equations consistent with a tabulated nuclear EOS. The results presented from a suite of 1D test problems demonstrate the superior performance of the characteristic limiter compared to the componentwise limiter. Moreover, after performing a parameter study on the limiter parameters, we found

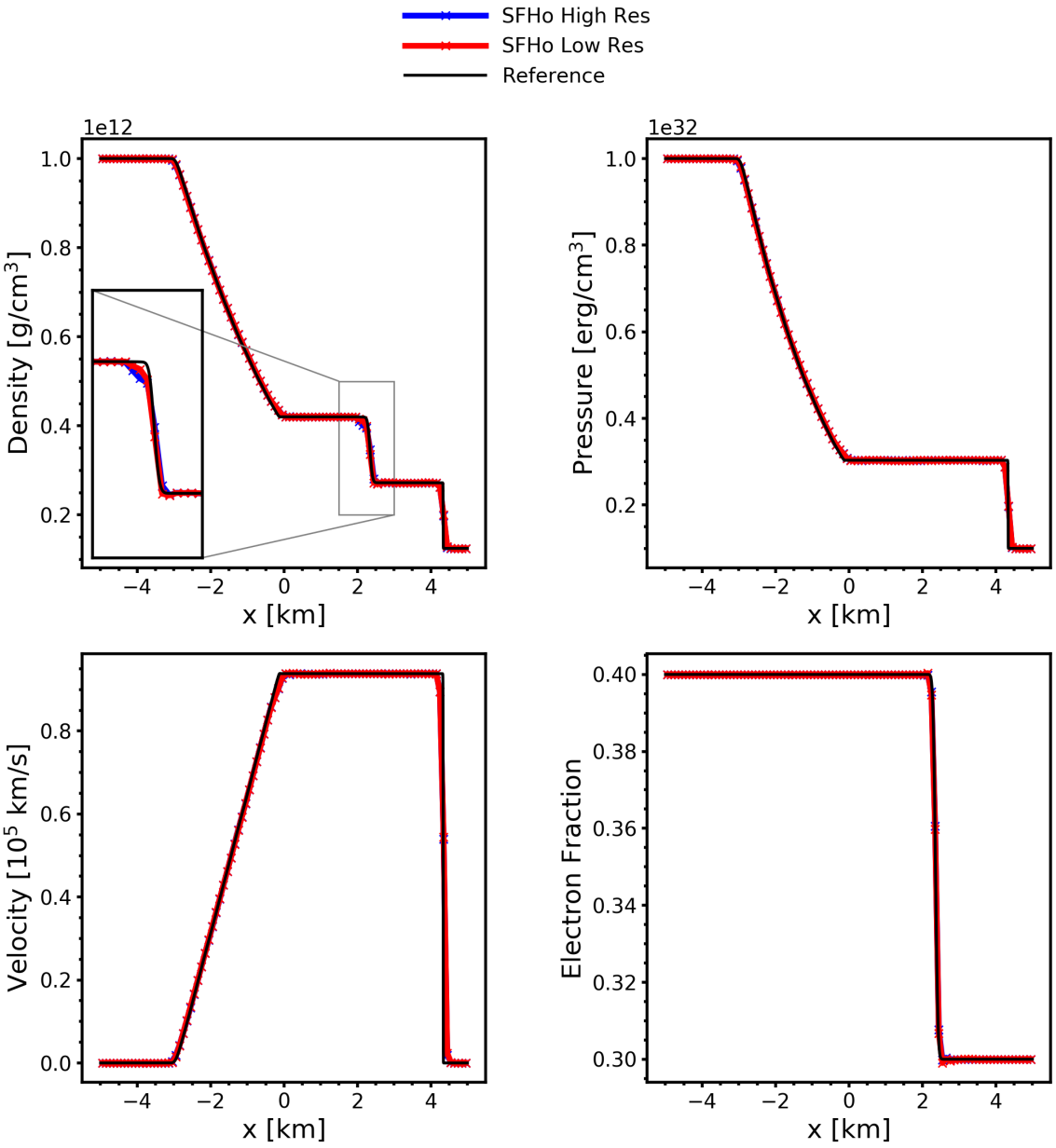

Figure 6. Comparison of numerical solutions to Sod's problem using 100 elements with characteristic limiting and two different EOS table resolutions (low: red, high: black) for density (upper left), pressure (upper right), velocity (lower left), and electron fraction (lower right) compared with a reference solution (black) using 10000 elements and the low resolution EOS table.

optimal limiting parameters for use with the Sod problem. While the optimal limiting parameters tend to be problem dependent, we hope to investigate if such an optimal parameter exists for CCSN applications. Tests with various EOS tables and table resolutions showed little to no sensitivity to the table or its resolution in the tests studied. For future work, we will extend these studies to other higher density regimes applicable to the CCSN environment. Planned near-future work on the nuclear equation of state compatibility of `thornado` includes the development of a positivity limiter and extension of the characteristic limtier to handle multi-dimensional, relativistic, and curvilinear problems. Because of the superior performance of characteristic limiting, we hope that this work will improve the fidelity of CCSN simulations.

Eirik Endeve and Anthony Mezzacappa acknowledge support from the NSF Gravitational Physics Program (NSF-GP 1505933 and 1806692).

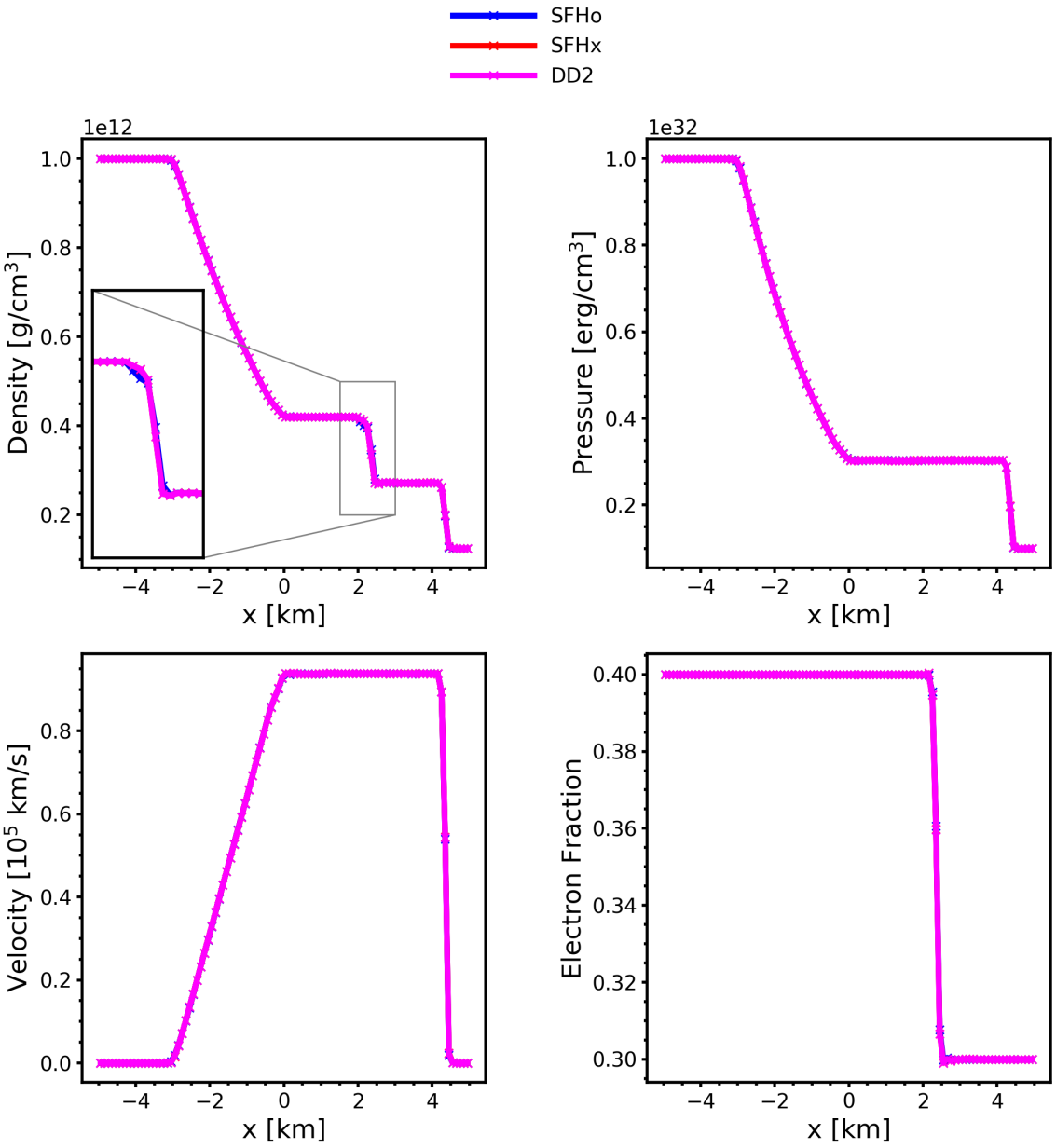

Figure 7. Comparison of numerical solutions to Sod's problem using 100 elements with characteristic limiting and three different EOS tables for density (upper left), pressure (upper right), velocity (lower left), and electron fraction (lower right).

Software: Matplotlib (Hunter 2007), NumPy (van der Walt et al. 2011), SciPy (Jones et al. 2001),

APPENDIX

A. DERIVATIVES

Derivatives of pressure with respect to τ, ϵ, and D_e in terms of the table variables ρ, T, y_e.

$$\left(\frac{\partial P}{\partial \epsilon}\right)_{\tau,D_e} = \left(\frac{\partial \epsilon}{\partial T}\right)^{-1}_{\rho,y_e} \left(\frac{\partial P}{\partial T}\right)_{\rho,y_e} \tag{A1}$$

$$\left(\frac{\partial P}{\partial D_e}\right)_{\tau,\epsilon} = \tau \left[\left(\frac{\partial P}{\partial y_e}\right)_{\rho,T} - \left(\frac{\partial \epsilon}{\partial y_e}\right)_{\rho,T} \left(\frac{\partial P}{\partial \epsilon}\right)_{\tau,D_e}\right] \tag{A2}$$

$$\left(\frac{\partial P}{\partial \tau}\right)_{\epsilon,D_e} = -\tau^{-2} \left[\left(\frac{\partial P}{\partial D_e}\right)_{\tau,\epsilon} (y_e - 1) + \left(\frac{\partial \epsilon}{\partial \rho}\right)_{y_e,T} \left(\frac{\partial P}{\partial \epsilon}\right)_{\tau,D_e}\right] \tag{A3}$$

REFERENCES

Bassi, F., Franchina, N., Ghidoni, A., & Rebay, S. 2013, International Journal for Numerical Methods in Fluids, 71, 1322, doi: 10.1002/fld.3713

Bethe, H. A. 1990, Reviews of Modern Physics, 62, 801, doi: 10.1103/RevModPhys.62.801

Blondin, J. M., & Mezzacappa, A. 2006, ApJ, 642, 401, doi: 10.1086/500817

Bruenn, S. W., Blondin, J. M., Hix, W. R., et al. 2018, ArXiv e-prints. https://arxiv.org/abs/1809.05608

Burrows, A. 2013, Reviews of Modern Physics, 85, 245, doi: 10.1103/RevModPhys.85.245

Cockburn, B. 2001, Journal of Computational and Applied Mathematics, 128, 187, doi: 10.1016/S0377-0427(00)00512-4

Cockburn, B., Lin, S.-Y., & Shu, C.-W. 1989, Journal of Computational Physics, 84, 90, doi: 10.1016/0021-9991(89)90183-6

Cockburn, B., & Shu, C. 2001, Journal of Scientific Computing, 16, 173

Cockburn, B., & Shu, C.-W. 1998, Journal of Computational Physics, 141, 199, doi: 10.1006/jcph.1998.5892

Colella, P., & Glaz, H. M. 1985, Journal of Computational Physics, 59, 264, doi: 10.1016/0021-9991(85)90146-9

Couch, S. M. 2017, Philosophical Transactions of the Royal Society of London Series A, 375, 20160271, doi: 10.1098/rsta.2016.0271

Couch, S. M., Graziani, C., & Flocke, N. 2013, ApJ, 778, 181, doi: 10.1088/0004-637X/778/2/181

Couch, S. M., & Ott, C. D. 2015, ApJ, 799, 5, doi: 10.1088/0004-637X/799/1/5

Dumbser, M., Guercilena, F., Köppel, S., Rezzolla, L., & Zanotti, O. 2018, Phys. Rev. D, 97, 084053, doi: 10.1103/PhysRevD.97.084053

Fu, G., & Shu, C.-W. 2017, Journal of Computational Physics, 347, 305 , doi: https://doi.org/10.1016/j.jcp.2017.06.046

Harten, A., Lax, P., & Leer, B. 1983, SIAM Review, 25, 35, doi: 10.1137/1025002

Hix, W. R., Lentz, E. J., Endeve, E., et al. 2014, AIP Advances, 4, 041013, doi: 10.1063/1.4870009

Hunter, J. D. 2007, Computing In Science & Engineering, 9, 90

Janka, H.-T., Hanke, F., Hüdepohl, L., et al. 2012, Progress of Theoretical and Experimental Physics, 2012, 01A309, doi: 10.1093/ptep/pts067

Janka, H.-T., Langanke, K., Marek, A., Martínez-Pinedo, G., & Müller, B. 2007, PhR, 442, 38, doi: 10.1016/j.physrep.2007.02.002

Janka, H.-T., Melson, T., & Summa, A. 2016, Annual Review of Nuclear and Particle Science, 66, 341, doi: 10.1146/annurev-nucl-102115-044747

Jones, E., Oliphant, T., Peterson, P., et al. 2001, SciPy: Open source scientific tools for Python. http://www.scipy.org/

LeVeque, R. 1992, Numerical Methods for Conservation Laws, Lectures in Mathematics ETH Zürich, Department of Mathematics Research Institute of Mathematics (Springer). https://books.google.com/books?id=3WhqLPcMdPsC

—. 2002, Finite Volume Methods for Hyperbolic Problems, Cambridge Texts in Applied Mathematics (Cambridge University Press). https://books.google.com/books?id=O_ZjpMSZiwoC

Mabanta, Q. A., & Murphy, J. W. 2018, ApJ, 856, 22, doi: 10.3847/1538-4357/aaaec7

Müller, B., Heger, A., Liptai, D., & Cameron, J. B. 2016, MNRAS, 460, 742, doi: 10.1093/mnras/stw1083

Müller, B., Janka, H.-T., & Marek, A. 2012, ApJ, 756, 84, doi: 10.1088/0004-637X/756/1/84

Murphy, J. W., Dolence, J. C., & Burrows, A. 2013, ApJ, 771, 52, doi: 10.1088/0004-637X/771/1/52

Murphy, J. W., & Meakin, C. 2011, ApJ, 742, 74, doi: 10.1088/0004-637X/742/2/74

O'Connor, E., & Couch, S. 2018, accepted to ApJ. https://arxiv.org/abs/1807.07579

Radice, D., Ott, C. D., Abdikamalov, E., et al. 2016, ApJ, 820, 76, doi: 10.3847/0004-637X/820/1/76

Radice, D., & Rezzolla, L. 2011, Phys. Rev. D, 84, 024010, doi: 10.1103/PhysRevD.84.024010

Schaal, K., Bauer, A., Chandrashekar, P., et al. 2015, MNRAS, 453, 4278, doi: 10.1093/mnras/stv1859

Schaal, K., & Springel, V. 2015, MNRAS, 446, 3992, doi: 10.1093/mnras/stu2386

Shu, C.-W., & Osher, S. 1988, Journal of Computational Physics, 77, 439, doi: 10.1016/0021-9991(88)90177-5

Sod, G. A. 1978, Journal of Computational Physics, 27, 1 , doi: https://doi.org/10.1016/0021-9991(78)90023-2

Steiner, A. W., Hempel, M., & Fischer, T. 2013, ApJ, 774, 17, doi: 10.1088/0004-637X/774/1/17

van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, Computing in Science Engineering, 13, 22, doi: 10.1109/MCSE.2011.37

Zanotti, O., Fambri, F., Dumbser, M., & Hidalgo, A. 2015, Computers and Fluids, 118, 204 , doi: https://doi.org/10.1016/j.compfluid.2015.06.020

Zhang, X., & Shu, C.-W. 2010, Journal of Computational Physics, 229, 8918 , doi: https://doi.org/10.1016/j.jcp.2010.08.016